2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Multi-Robot Search for a Moving Target: Integrating World Modeling, Task Assignment and Context

Francesco Riccio[1], Emanuele Borzi[1], Guglielmo Gemignani[1], Daniele Nardi[1]

Abstract— **In this paper, we address coordination within a team of cooperative autonomous robots that need to accomplish a common goal. Our survey of the vast literature on the subject highlights two directions to further improve the performance of a multi-robot team. In particular, in a dynamic environment, coordination needs to be adapted to the different situations at hand (for example, when there is a dramatic loss of performance due to unreliable communication network). To this end, we contribute a novel approach for coordinating robots. Such an approach allows a robotic team to exploit environmental knowledge to adapt to various circumstances encountered, enhancing its overall performance. This result is achieved by dynamically adapting the underlying task assignment and distributed world representation, based on the current state of the environment. We demonstrate the effectiveness of our coordination system by applying it to the problem of locating a moving, non-adversarial target. In particular, we report on experiments carried out with a team of humanoid robots in a soccer scenario and a team of mobile bases in an office environment.**

I. INTRODUCTION

Multi-Robot System (MRS) have been deeply studied during the past few years to develop effective solutions for multi-robot task execution, distributed world representation, and robust coordination. Such techniques for Multi-Robot Systems have shown to successfully handle several requirements in different environmental setups. Moreover, proposed approaches have been evaluated against varying conditions (e.g. communication bandwidth) and shown to be scalable and possibly adaptive. However, there are situations where the changes in the scenario make it necessary to dramatically change the coordination strategy. The initial motivating example of our research was the coordinated search for the ball in a robotic soccer scenario. When none of the team members perceives the ball, the strategy to search the ball changes significantly depending on whether there was a referee call sanctioning that the ball exited the field or not. This phenomenon is not specific to soccer, but it can be found in many other deployments of multi-robot teams. For example, when locating a target for robotic logistics, tracking a lost target in surveillance applications, search and rescue scenarios, and for service robots operating in indoor environments. We propose a novel approach to multi-robot search integrating world modeling, task assignment and context to address the aforementioned application domains. In fact, we have generalized the approach of searching for the ball to a generic target location, which is dependent on the specific circumstances that are detected during operation.

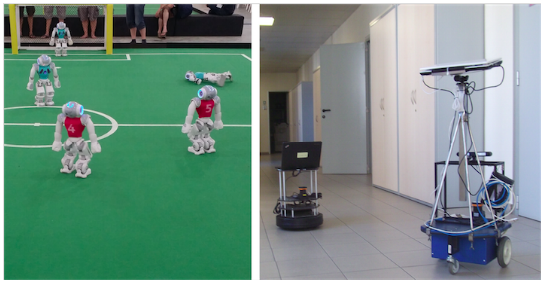

Fig. 1. Picture of the Turtlebot, Erratic and Nao robots coordinated with the proposed approach.

Our analysis of the literature shows that the proposed approaches to coordination are developed along two main directions. On the one hand, there is a significant body of work that can be characterized as *distributed world modeling*, where the aim is to share information so that each robot can make decisions on which action to take, based on a world model that is built through the exchange of the local views of each robot in the team. On the other hand, methods such as *distributed task assignment* exchange information (i.e. utilities) to allow each robot to choose the task that is most appropriate, considering the preferences of the teammates.

While there are sometimes applications where sharing the world model is not advisable, either because of communication constraints or because of implementation requirements, a suitable approach to distributed world modeling can substantially improve the coordination via task assignment. Hence, we design a distributed world model, that is specific to the chosen application domain, and is updated based on the information received by the teammates. In order to avoid heavy requirements on the network, our distributed world model relies upon an abstract representation of the environment and a limited exchange of information. Taking into account the distributed world model, the system assigns the set of active tasks and coordinates the robots via distributed task assignment. One novel contribution of our proposal is the integration of distributed world modeling within a task assignment framework.

In addition, in order to handle the changes in the environment that might require a complete change in the strategy, we categorize any identifiable configuration of the environment as *Contextual Knowledge*. More specifically, we allow robots to detect *events* which are used to determine *contexts*. Such events can be received from the network, or perceived by

[1] Dept. of Computer, Control, and Management Engineering, Sapienza University of Rome, via Ariosto 25, 00185, Rome, Italy {riccio,borzi,gemignani,nardi}@dis.uniroma1.it

978-1-5090-3761-2/16/$31.00 ©2016 IEEE

the robots. Then, we exploit the high level representation of their distributed world model to encode contextual information, and adapt the team strategy dynamically. We represent context as a combination of two components, namely *environmental* and *task-related* knowledge [1]. Contexts can be triggered by specific events related to the operational environment and support the selection of the best team strategy. A second distinguishing feature of the proposed approach, is the explicit modeling of contexts which allows for a context switching depending on the environment monitoring (e.g. network bandwidth)

Summarizing, the proposed approach contributes in 1) combining a dynamic distributed world model with a distributed market-based techniques for role assignment, 2) integrating Contextual Knowledge in a multi-robot system to improve the overall performance of the team. We successfully deploy our system in two different case studies for locating a moving, non-adversarial target: we consider the problem of locating an unseen ball in a RoboCup soccer game and the problem of finding a moving person in an office environment. Our contribution has been deployed on several simulated and real robots, including a team of Turtlebot and Erratic robots and a team of humanoid NAOs. Fig. 1 shows a picture of the robots coordinated with our proposed technique. We carried out a substantial set of experiments that show the effectiveness of our approach.

In the remainder of the paper, we first present an overview of related work, focusing on past research in multi-robot coordination and context aware systems. We then provide an overview to our context-aware coordination describing all of our contributions thoroughly. Then, we present two applications of the approach to two case studies of non-adversarial target search. Lastly, we conclude with a discussion of the approach and remarks on future work.

II. RELATED WORK

A reliable coordination module is the core component of systems where multiple robot units need to cooperate to achieve a common goal. During the last years, the approaches to Multi-Robot Systems have been noticed and categorized in different survey papers. For instance, Cao et al. [2] give a first categorization to multi-robot coordination systems. Dudeck et al. [3] provide a new taxonomy based on communication and computation aspects. Moreover, Parker [4] highlights the issues and research topics related to MRS systems, while Stone and Veloso [5] discuss the relation of MRS and the field of Multi-Agent System. Finally, in [6] the authors provide a classification of multi-robot approaches focusing on coordination issues of a MRS. Our survey of the literature (that we summarize here due to lack of space) highlights differences in current MRS approaches, and we propose a new point of view categorizing existing work by their assumptions on *Distributed World Modeling* and *Distributed Task Assignment*. The approaches to coordinate MRS are manifold and typically depend on the goal of the application that the robotic system is aiming to. We refer at *distributed task allocation* (DTA), when the application is focusing on generating and optimizing the coordination criteria governing the team of robots. While, we refer at *distributed world model reconstruction* (DWM), when the focus is on exchanging information that allows for building a global model of the world that integrates information that cannot be acquired locally by each robot (e.g. reconstructing a map of the environment).

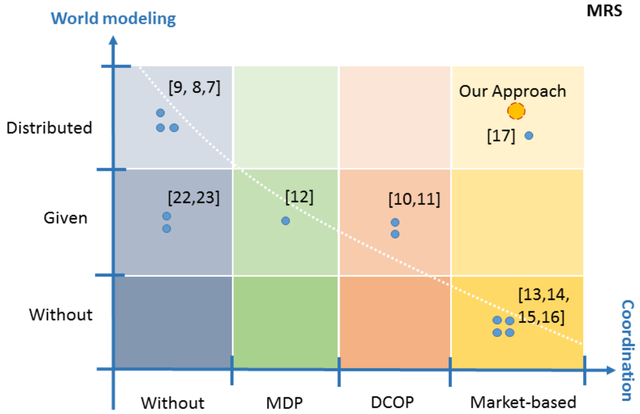

Fig. 2. Related work at a glance. The figure illustrates how our contribution compares to existing approaches. Such approaches are represented with dots that, if in the same area, represent contributions with similar properties. On the y-axis we categorize MRS depending on their assumptions on DWM, while on the x-axis on the technique used to coordination. The white-dotted line highlights how the majority of approaches in MRS place themselves with respect the proposed taxonomy. Our proposed method combines both a dynamic representation of the world and a market-based coordination approach (better view in color).

To highlight how our contribution compares to existing approaches, we categorize existing works on Fig. 2 by considering their assumptions on a distributed world model on the y-axis, and their coordination approaches on the x-axis. The representation used is qualitative and serves to give an idea of the current trend in multi-robot systems. Specifically, we place current literature on the y-axis by differentiating three different modalities of formalizing the distributed world model: a *distributed* approach, a world representation, which is usually given and static during the evolution of the team mission, and finally, the third category, where the world model is explicitly represented unchanged. Conversely, on the x-axis the coordination approaches are organized in four categories: without, the robots do not have a predefined method for collaborating with teammates; coordination based on Distributed Markov Decision Processes, where the robot formalize a unique action policy for all the team; approaches based on Distributed Constraint Optimization, where the robots minimize an objective function; and finally, market-based coordination algorithm, that currently represent the set of methodologies that less rely on the robots' internal representation of the world.

Approaches to distributed world modeling, typically rely on a metric representation of the surrounding scenario. For example, Zhou et al. [7] match relative reconstructed maps with an EKF-based SLAM approach. Howards [8] employs particle filters to merge several maps carried out by each unit in the team. As DWM approaches, we also consider Multi-robot localization as in [9], where robots update their world state through mutual robot detection. The majority of these works exploit a distributed world modeling of the surrounding environment by reconstructing the scenario through 2D or 3D maps. We also rely on a distributed world model, but we adopt a more abstract representation of the environment by exploiting a distributed topological representation of the world, thus offering a much lighter and generalizable level of environmental representation.

The problem of *Distributed Task Allocation* is expressed as the problem of relating a set of tasks to a set of robots. Many of the works that can be found in the literature consider the world model as "given" and suitably represented to run and evaluate the coordination algorithm. For instance, both in Okamoto et al. [10] and Correa et al. [11], the authors use distributed constraint optimization approaches to coordinate a team of robots in simulated grid world; Capitan *et al.* [12] formalize decentralized POMDP based on auctions in order to perform cooperative surveillance. Market-based techniques are a well established approaches to optimize coordination algorithm, even without an explicit formalization of the surrounding world. Gerkey and Matarić [13] address the problem of heterogeneous robot cooperation by employing a publish/subscribe system on task bidding. Dias and Stentz in [14], instead, enable a team of robots to bid for a given set of tasks in a fully distributed way. The team of robots self-organizes in sub-groups and bids for resources. More recently, Luo *et al.* [15] introduce an iterative greedy auction algorithm to allocate task among the team. MacAlpine *et al.* [16] adopt market-based utility estimation to coordinate a team of robots in a RoboCup scenario. We also rely upon market-based utility to generalize our system, but we combine it with Distributed World Modeling to select a suitable strategy and perform a better assignment. Few approaches explicitly formalize separately the DWM and the DTA module. For example, in [17] the authors use the metric representation of an environment to assign different roles in an exploration task. However, their centralized coordination system does not adapt to environmental constraints (e.g. network issues), or encodes environmental knowledge at an higher representation level. We exploit the DWM high level representation to select the best strategy to be executed. Then, we modify the set of assignable roles; the coordination parameters; and we encode strategies in different utility functions in order to maximize the effectiveness of the system in accordance with the situation at hand. To achieve such a flexibility, we formalize environmental and task-related information as *Contextual Knowledge*. Then, we enable the robots to recognize configurations of the environment as *context* and respond to them adaptively.

Multiple works address the problem of leveraging the contextual knowledge of the operational scenario. For instance, Robotic systems can exploit contextual knowledge to develop a context-aware indoor system [18]; contextualize tasks to satisfy particular requirements during the execution [19]; and exploit "introspective agent knowledge" to design agent behaviors et al. [20]. Contextual knowledge has also been exploited in the framework of MRS. For instance, Kaminka's BITE system [21] propose a framework to coordinate a team of robots in area coverage tasks or team formation maintenance. However, in their work, contextual knowledge is strictly related to the running tasks and it does not extend to the formalization of environmental knowledge (e.g. network evaluation). In particular, the authors do not represent dynamic changes of the operational environment to adapt accordingly. In this work, we explicitly evaluate the current environmental configuration and the robots' procedural knowledge to specify the robots behavior to the current situation. Contextual knowledge, in fact, enables robots to reason and understanding in a more *natural* way about the surrounding dynamic environment and respond to it. As shown in Sec. V, this *contextual knowledge* considerably improves the system performance in both our application.

Summarizing, we propose a new approach to team coordination. Our system builds an abstract (and thus light) Distributed World Model to *synchronize* robots. Then, it detects a strategy by automatically formalizing contexts, and finally, it performs a Distributed Task Assignment with respect to the chosen strategy and the current representation of the DWM in order to associate robots and tasks.

III. APPROACH

In this section we present our approach to multi-robot coordination combining *Distributed Task Assignment* and *Distributed World Modeling*. The idea is to simultaneously exploit the robustness of DWM approach with the efficiency of DTA techniques. The coordination approach is then enhanced by leveraging the contextual-knowledge of the scenario. In this section, we provide a formal setting which embodies all these elements.

A. Combined Coordination Model

The core of our coordination algorithm relies on a *Distributed Task Assignment* (DTA), based on *utility estimations*. DTA is an example of market-based coordination: at each step, the algorithm evaluates the possibility of a given robot to perform a given task according to its *utility value*. Given a set of M tasks $T = \{\tau_1, \tau_2, ..., \tau_m\}$ for a team of N robots $R = \{r_1, r_2, ..., r_n\}$, a *utility estimation vector* (*UEV*) can be seen as the estimation of "how good" a particular robot is for each task τ_i at a given time t. If one denotes with $b_{i,j}(t)$ the estimation that the robot r_i computes for the task τ_j at time t, its *UEV* can be expressed as:

$$UEV_i(t) = \left[b_{(i,1)}(t), \quad \dots \quad , b_{(i,m)}(t)\right] \tag{1}$$

The *utility estimation matrix* (*UEM*) represents the utilities of all the members in the team, since each row i is the *UEV* corresponding to each robot r_i. This matrix is computed individually by each robot and it is built by gathering the vectors sent by the teammates and transposing them:

$$UEM_i(t) = \left[UEV_1(t), \quad \dots \quad , UEV_n(t)\right]^T . \tag{2}$$

Given such a representation, the task assignment can be computed by evaluating each task of the $UEM(t)$ columnwise. Thus, by considering the scores of each robot r_i, we assign to a given task τ_j the robot with the highest score $b_{(i,j)}(t)$. This mapping is performed by the *coordination mapping function* Φ_i of the robot i-*th* :

$$\Phi_i : R \rightarrow T. \tag{3}$$

Since we also want to exploit the high level formalization of the surrounding scenario, we allow the *UEM* to be modified based on a distributed world model. In order to build a reliable distributed representation of the current world state, we need to synchronize our robots on a common representation of the environment and keep it updated over time. Hence, we define our *distributed world model* (*DWM*), which represents the knowledge of the world reconstructed from a set of partial local models. Precisely, we refer to the DWM_i as the distributed world model locally reconstructed by the robot i-*th* . In our case studies a robot's local model represents the probability of finding the target in a specific part of the environment, estimated through its limited and partial knowledge of the scenario.

Given a set of robots $R = \{r_1, r_2, ..., r_n\}$ and their corresponding local models $LM_j(t)$ at a given time t, we are able to *reconstruct* the distributed world model for the robot j-*th* as

$$DWM_j(t) = f(LM, t) \tag{4}$$

where f is a specific distributed model update function and LM represents the set of all the local models $\{LM_j\}_{j=1}^{N}$. The function f needs to be specified for each case scenario, as we will see in the next section.

Any distributed approach needs to rely on a robust algorithm for reconstructing the *DWM*. We adopt an *event-based system* to efficiently manage the robot internal representations of the world. This system is based on the concept of *events*. These events can be either sensed by a single robot or they can be told by an external agent to the entire team. For example, an event may be represented by a door being opened or by a person telling the team that the target was previously seen in a particular location of the environment. We use these events to change the world representation. Formally, we define *model update* a function Ψ that takes in input an event $e \in E$ and a local model $LM_j(t)$ and outputs a modified local representation of the world. Thus, the local model of each robot will be modified in the following way:

$$\overline{LM}_j(t) = \Psi(e(t), LM_j(t)). \tag{5}$$

When the model update function and the *event system* are implemented in the same way on all the robots, they can share only local events, instead of communicating their entire local model. This considerably reduces the communication overhead. The algorithmic formalization of the distributed model update and dynamic task assignment is shown at the end of the next section.

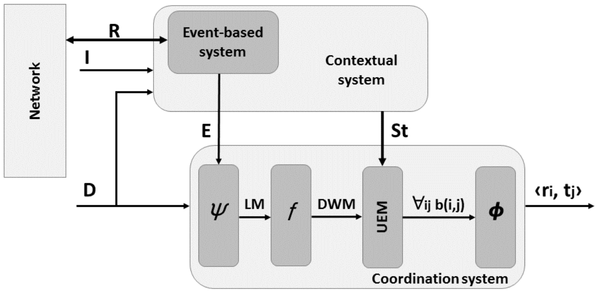

Fig. 3. Sketch of our framework. The contextual system informs the team about the current context formalizing the most suitable strategy. The coordination system coordinates the robots based on the contextual information: it first updates the local models through Ψ, then reconstructs the distributed world model through f. Finally, it computes the UEM outputting a mapping from robots to tasks.

B. Context System

Contextual knowledge can increase the robot performance in the tasks to be accomplished. By detecting the context we can dynamically change the team strategy. In our approach, we use these contexts to allow the robots to represent situations that require a different coordination strategy. We introduce a *Context System* which gathers contextual knowledge and outputs the best strategy to adopt during the execution of the team mission. Such strategy carries out high-level information directly represented as parameters of the underlying coordination system. These parameters are used to select the proper strategy. Formally, we characterize such a system as a function *CS* which takes as input the set of sensory data *D* and external input events *I*. *CS* generates a coordination strategy *St* for each different context:

$$CS : \ [\ D \ \times \ I\ ] \rightarrow St \tag{6}$$

The information flow of our framework is shown in Fig. 3. The context system influences the regular execution of the coordination system by means of different *strategies*. More formally, at any given time t, the coordination strategy influences the mapping function expressed by Eq. 3 by modifying the structure of the utility matrix and/or the utility estimations of the $UEM_i(t)$ (Eq. 2). We refer as $\overline{UEM_i}$ to the utility matrix which has been dynamically adapted to the current strategy.

As shown in the following section, the adaptation of the utility function is extremely effective when the requirements of operational scenario are substantially changing.

Algorithm 1 reports the overall coordination protocol locally run by the single robot. Specifically, the robot i-*th* detects the current context by evaluating data sources and external input events (line 2). Then, it retrieves the information needed from the other teammates (line 4) to locally reconstruct their local models (line 6). The robot is now able to estimate a global model DWM_i by considering the reconstructed local models $\overline{LM_j}$ of each connected teammate (line 8). At this point, the robot can compute the set of reference tasks according to the current state of the world (line 10) and its utility vector UEV_i (line 12). The robot multicasts its utility vector, and waits for the estimations of the other teammates (line 16) Finally, by means of Eq. 2, the robot calculates the utility matrix $\overline{UEM_i}$ (line 19); Then, the procedure locally computes and returns the most suitable task for the robot i-*th*.

Algorithm 1: Context-Coordination

```
Input: sensory data $D$, input events $I$, teammates $R$
Data: set of local models $LM$, reconstructed distributed world
      model $DWM$, context system $CS$, teammate $j$ state $TS_j$,
      Team Strategy $St$, set of Task to assign $T$, robot $i$
      utility estimation vector $UEV_i$, teammate $j$ utility
      estimation vector $UEV_j$, utility estimation matrix $\overline{UEM_i}$
Output: Task for the robot i-th $T_i$
1  begin
2      // Update context knowledge
3      $St \leftarrow$ updateContext(CS, I, D)
4      // For each teammate $j$ receive its updated state
5      $\{TS_j\}_{j=1}^{N} \leftarrow getTeammateUpdate(R)$
6      // Update teammates local models $LM_j$
7      $\{\overline{LM}_j\}_{j=1}^{N} \leftarrow$ updateLM($\{TS_j\}_{j=1}^{N}$)
8      // Update the distributed world model for the robot i-th
9      $DWM_i \leftarrow$ reconstructDWM($\{\overline{LM}_j\}_{j=1}^{N}$)
10     // Compute Tasks
11     $T \leftarrow$ computeTasks($DWM_i$)
12     // Compute the utility vector
13     $UEV_i \leftarrow$ computeUEV($T$, $St$)
14     // send the utility vector
15     sendUEV($UEV_i$)
16     // For each teammate $j$ receive utility estimation by other
       teammates
17     $\{UEV_j\}_{j=1}^{N} \leftarrow getUtilityVectors(R)$
18     // Compute the utility matrix
19     $\overline{UEM_i} \leftarrow$ computeUEM($\{UEV_j\}_{j=1}^{N}$)
20     // Select the task according to the utility matrix
21     return $T_i \leftarrow$ mapping($\overline{UEM_i}$)
22 end
```

IV. APPLICATION SCENARIOS

To prove the effectiveness of our approach, we address two settings: a soccer game during which the robots need to search for a moving ball, and an office environment, where a team needs to locate a person in a non-adversarial setting. Our algorithm can be downloaded and installed[1][2]. These two settings have also been used to quantitatively evaluate our contribution.

A. Soccer Case Study

Our approach to coordination was initially motivated and developed in RoboCup soccer games. In fact, it has been first deployed on a team of NAOs. NAOs are commercial, autonomous, 25-DOF humanoid robots. They are equipped with a wide variety of sensors and actuators, including two CMOS cameras, multiple proximity sensors, four microphones, and two speakers.

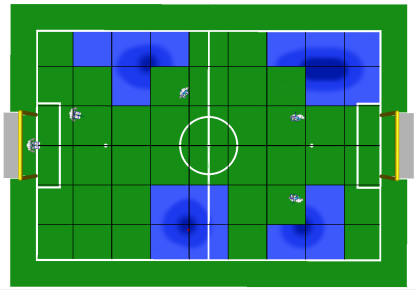

Fig. 4. Distributed World Model for the soccer scenario (on the left) adn the .

In this setting, a team of robots plays in a 9*x*6 *meters* soccer field of the RoboCup Standard Platform League. In our coordination algorithm this field is represented as an occupancy grid. Each cell in this grid features a score, representing how likely it is to find the ball inside it. Fig. 4 shows the DWM reconstructed by a team of NAOs in a simulated environment.

1) Search for the Ball: When the team does not see the ball a collaborative search task is needed. Specifically, the Context System is able to recognize two *task-related contexts*: *Throw-In* and *Ball Lost*. The *Throw-In* takes into account the setting in which the ball has rolled out of the field and the robots are not able to see it any more. Such a context is recognized considering the single perceptions of the robots and on the messages sent to the whole team by the external Game Controller [3] Instead, *Ball Lost* takes into account the situation in which the game is regularly played, but all the players in the team have lost track of the ball. This particular context is recognized only through the perceptions of the robots.

When a context is recognized the robots start coordinating and sharing information. In this setting the robots share only the outcome of two actions as events, namely *clear area* and *ball found*, which are associated with the centroid of a visited area. Specifically, such events are locally detected by each robot. Upon detection the agent sends a message to the teammates, which when received it is used to update the distributed model of the robots according to the event type (Eq. 5). For instance, a “ball clear” event has the effect of reducing the probability of finding the ball in a given area. In particular, while searching for the ball, the robots exchange the centroids of controlled areas. When the robot i-*th* receives an event messages by the robot j-*th* and updates its local, then it merges the new information in the global distributed model DWM. To this end, Eq. 7 shows how the reconstruction function f (Eq. 4) is implemented in this case. This function is defined as the union of the score of each cell, updated as:

$$DWM_i \;:\; \forall x,y\ cell_i^{\langle x,y\rangle} = \underset{cell_j \in LM_j}{\arg\min}\,\{score(cell_j^{\langle x,y\rangle})\} \qquad (7)$$

where the score of the $\langle x,y\rangle$ cell in the overall representation of the robot i-*th* is the minimum score among all the local

[1] `https://github.com/francescoriccio/RCoordination`

[2] The coordination system can be found integrated in the soccer specific framework at `https://github.com/SPQRTeam/spqrnao2015`

[3] The Game controller is an external electronic referee used to communicate with the playing robots during a regular match.

models LM_j of each robot. Intuitively, it informs the i-*th* that one of its teammate has recently controlled a given area and the search can be directed elsewhere.

When the events are received from the team, the Context System determines the current context outputting a set of contextual parameters C used to leverage the team strategy. In this specific scenario, contextual parameters are used to activate the set of tasks that are related to the detected context. For example, let us consider the setting in which the robots are searching for the ball. If the *Game Controller* notifies that the ball is rolled out of the field, then the team can temporarily assume that the most promising area to look at are represented by the long sides of the game field and assign the set of more specialized tasks related to the *Throw-In* context. Conversely, if such a signal is not received, then the search strategy cannot be specialized to particular areas and a complete field coverage is needed.

In this setting we associate the team of robots with a set of roles based on utility estimations and in accordance with Figure 2. Specifically, in the "Ball Lost" context, the utility estimations are based on the position of the robots and the location of the cell *i-th* to explore. With this approach, the team of robots coordinate in a context-aware fashion. Thus, as we will see in the next section, the team of soccer robots drastically increases its performance in finding the ball.

2) Network Monitoring: In real applications, the robot communication is one of the main problems and it is not typically monitored at any time to adopt the coordination. In RoboCup, network reliability is one the main issues to be tackled in order to design a workable coordination framework. In this perspective, multiple proposed approaches evaluate their performance against unstable network conditions and limited band-width. Several solutions adjust the team displacement in order to maintain connectivity [22] or periodically share information to bound the network overload [23]. However, to the best of our knowledge, there is no approach that continuously performs an online analysis of the network bandwidth to select the most appropriate coordination strategy according to the current network performances. In both our case studies, the network condition is detected directly by the team of robots by evaluating the *Round Trip Time (RTT)* of the packages shared among the team. Such evaluation is useful to detect the current network context that can assume three different values, namely *no network*, *unreliable network* and *reliable network*. For example, in the soccer scenario, we modify the team formation (set of roles) of the team in accordance to the network communication level.

B. Office Case Study

In this subsection, we instantiate our framework to the problem of multi-robot target localization by applying our system to different operational scenarios. Several works address the problem of pursuit evasion. Here, we focus on non-adversarial target localization, as in Hollinger et al. [24]. The authors propose a system to address the problem of *Multi-Robot Efficient Search Path Planning*, through an approximation algorithm based on finite-horizon planning and implicit coordination. The authors deploy their system in multi-robot search in underwater and rescue scenarios. Similarly, Geyer et al. [25] perform a target search in a urban environment through search trees and particle filters. However, these solutions do not formalize a distributed world model which, is given to the team before operation and, remains known and static. Most importantly, none of the existing approaches adapt the searching strategy depending on the current world state.

Here, we consider a complex setting, where the robots in the team coordinate to find a person in a given map. In this setting, we choose to discretize the office environment through a *topological graph*, as in [24], thus showing the effectiveness of the approach in a completely different representation of the environment.

The Context System is implemented as a search on a decision tree and a knowledge base is used to recognize the set of contexts $\{Meeting, Lunch, Morning, Afternoon\}$. This knowledge base includes information about the scenario, such as scheduled meetings, habits, or room and object positions. In this case, we exploit contexts to assign different initial scores to the most promising nodes to look at, and influence the search accordingly. To this end, we semantically label the environment where the robots operate. In this way, we are able to perform spatial reasoning about objects and rooms, which helps carrying out the task.

In this case, even though the set of *task-related context* are determined by the daytime or daily meetings, the set of action outcomes to share among the teammates is wider due to the more complex nature of the environment. In this scenario, the robots share the following events *target near location*, *door opened*, *door closed*, *clear area*, and *person found*. Specifically, *target near location* is multicasted if one of the robots is informed that the target has been seen near a particular location. Instead, *door opened* and *door closed* are communicated whether a robot perceived the status of a door has changed. Finally, *clear area* and *person found* are respectively shared when a node has been visited and when the target is found. These information are associated with a set of nodes of the topological graph. Thus, the team can reconstruct the DWM by applying:

$$DWM_i \ : \ \forall n \ node_i^n = \operatorname*{arg\,min}_{node_j \in LM_j} \{score(node_j^n)\} \qquad (8)$$

which states that for each node n in the distribute world model of the i-*th*, the associated score is the minimum found in all the local models.

According to Equation 2, we coordinate the robots based on their utility values, with respect to a given set of *most-likely* nodes. The utility score of each pair $\langle r_i, \tau_j \rangle$ (i.e. $\langle robot_i, node_j \rangle$) is computed according to the cost of the path connecting the robot *i-th* and the node *j-th*. We use the Dijkstra algorithm for searching the optimal path p^* between two given nodes and to evaluate the best mapping from robot to nodes. In this case, contexts are used to prioritize the areas to look at according to the scheduling of the searched subject.

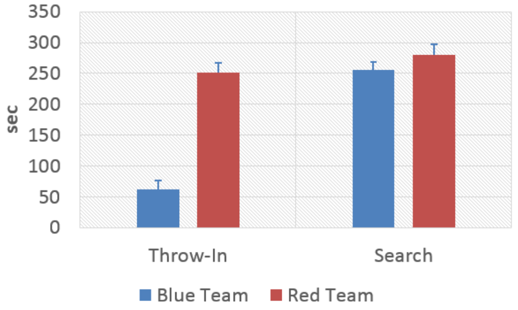

Fig. 5. Cumulative time during which the ball was not seen in a 10 minutes game for the two contexts Throw-In and *Ball lost*. The results were averaged over 100 runs.

For instance, if the team is within a *meeting context*, then the nodes that are associated to the person's office are reranked as the most promising areas to look at.

V. EXPERIMENTAL RESULTS

In this section we describe the analysis carried out in the soccer and office scenarios. The reported results have been obtained by extensively running our system in simulation, before executing it on real robots.

A. Results in the Soccer Scenario

The virtual environment where experiments are carried out is part of the *B-Human architecture*, which provides a RoboCup-dedicated simulation platform entirely written in C++[4], that features a rather accurate model of the behavior and capabilities of the humanoid robot in the field. In the soccer case study, our goal is to cooperatively search for the ball in highly changing environment. Accordingly, in order to show the effectiveness of our system, we first compared our context-aware coordination against a team that cannot distinguish between a *Throw-in* and a *Ball Lost* context. Both teams implement the same distributed world model (DWM), and share the same combined coordination model; however, the red team is not equipped with a contextual system. We measured the cumulative time during which the ball was not seen by the team in a game (i.e., 10 minutes). Fig. 5 reports the results averaged over 100 runs for the two different contexts considered.

Our algorithm was able to recognize the contexts and specialize the search, thus resulting in an overall better performance. It is worth noticing the effects of the different level of information available in the two contexts. In fact, when more detailed information was provided (e.g., in the *Throw-In* context), an increase in performance is noticeable. Instead, in both contexts the red team was not able to exploit the available information, always performing an *uninformed search* in all of the different contexts.

In a second experiment we measured different strategies while varying the reliability of network communication. More specifically, we developed an external tool for artificially introducing network delays in the simulations. We allow our team to detect the *unreliable network context* when the delays were above a certain thresholds. As in the previous test, in this setting the red team implements the same underlying coordination system, but, it is not able to detect network contexts. In this scenario, we adjust the strategy of our team to have a more aggressive formation, send a restricted number of packages to reduce the network overload of the team, and assume more static behaviors in order to reduce errors in the transmission of processed data. As we do not gave a specific *task to test*, we can only verify the quality of the role assignment by considering the scores of the games. Table I reports the results obtained in 173 runs of the experiments.

TABLE I

GAME RESULTS OF THE BLUE TEAM OVER 173 RUNS OF A SOCCER MATCH (I.E. 10 MINUTES).

	wins	losts	ties	games
blue	95	36	42	173

The results show a considerable difference in the number of won matches for the blue team that were able to categorize network contexts and adjust its coordination strategy dynamically. The scores prove that an adaptation on the operational scenario is always preferable when possible. Our coordination algorithm has been firstly validated on extensive testing sessions, and then implemented on real NAOs to allow the team of robots to compete during RoboCup matches. To the best of our knowledge, this is the first example of coordination that is adapted by a continuous monitoring of the network performance.

B. Results in the Office Scenario

The office experiments are carried out in the STAGE simulator by implementing our coordination system within the ROS framework. In this case, a team of mobile bases had to search for a person in the office environment. In this setting, we varied the number of robots performing the search, comparing our approach with other algorithms for exploration. Specifically, we compared it with two search strategies: a *random walk*, where the robots randomly explore the environment without coordinating; and a *coverage* search, where the team uses a DTA to coordinate, by keeping track of the visited nodes and randomly choosing the next ones to be explored. Fig. 6 illustrates the average time in seconds needed to find the moving target. This measure has been averaged over 10 runs for each configuration. The experiment was recorded as a failure, if the team needed more than 300s to complete the search.

The results of the experiments reported in Fig. 6 show that with our approach the performance improves as the number of robots grows and the information shared increases. Indeed, contextual information helps to properly evaluate the dynamics of the environment and rerank the areas to search next according to the current context. The results confirm our hypothesis, as the average time in locating the target considerably decreases, when context are properly formalized. Overall, our approach performed better in all of the considered configurations in terms of both time needed to complete the algorithm and percentage of successful tasks.

[4] https://www.b-human.de

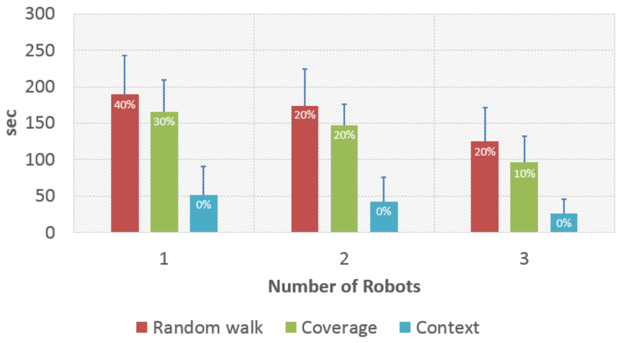

Fig. 6. Average time needed to locate the target for the considered algorithms. The percentages represent the ratio of failed tasks.

VI. Conclusion

In this paper, we considered the problem of coordinating a team of autonomous robots when capable of handling contextual knowledge of the scenario they operate in. To this end, we contributed an approach that allows for a distributed modeling of the environment, and adapts the coordination system to dynamic changes of the scenario. Accordingly, we exploit contextual knowledge to categorize environmental configurations and improve the team effectiveness. Our coordination algorithm has been applied to the problem of locating a moving, non-adversarial target in two different settings. We successfully deployed our coordination system on multiple robots: specifically, our experiments report indetail the performance of our contribution on a team of NAO robots in a soccer scenario and on a team of mobile bases in an office environment. In both scenarios, we found a significant reduction in the time needed to find the target, underlining the effectiveness of the approach. More specifically, as opposed to previous work aiming at developing methods that can scale up with respect to varying factors (e.g. communication bandwidth, delays), we propose an approach where the system can handle the changes in the operational scenario and select the best strategy online.

In the presented work, we have focused on the formalization and implementation of the framework, providing few example of the context detection and distributed world modeling. As a future work we are investigating the problem of representing contexts that need multiple and nondeterministic perceptions to be recognized, and to allow the team of robots to handle situations where the construction of the world model became more challenging. In fact, in the proposed scenarios, the coordination system assumes both contexts and events as pre-defined by an expert user. However, in unknown and unstructured environments they cannot always foreseen. To overcome this issue we want to investigate methodologies to discover context and events during robot mission by adapting the *Context System* to the current scenario and robot mission.

References

[1] R. M. Turner, "Context-mediated behavior for intelligent agents," *International Journal of Human-Computer Studies*, vol. 48, no. 3, pp. 307–330, 1998.

[2] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous robots*, vol. 4, no. 1, pp. 7–27, 1997.

[3] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.

[4] L. E. Parker, "Current state of the art in distributed autonomous mobile robotics," in *Distributed Autonomous Robotic Systems 4*. Springer, 2000, pp. 3–12.

[5] P. Stone and M. Veloso, "Robot teams: From diversity to polymorphism," *A Survey of Multiagent and Multirobot System*, 2002.

[6] A. Farinelli, L. Iocchi, and D. Nardi, "Multirobot systems: a classification focused on coordination," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 5, pp. 2015–2028, 2004.

[7] X. S. Zhou and S. Roumeliotis, "Multi-robot slam with unknown initial correspondence: The robot rendezvous case," in *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006, pp. 1785–1792.

[8] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," vol. 25, no. 12, pp. 1243–1256, 2006.

[9] S. Roumeliotis and G. Bekey, "Distributed multi-robot localization," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 781–795, Oct. 2002.

[10] S. Okamoto, N. Brooks, S. Owens, K. Sycara, and P. Scerri, "Allocating spatially distributed tasks in large, dynamic robot teams." Int. Foundation for Autonomous Agents and Multiagent Systems, 2011, pp. 1245–1246.

[11] A. Corrêa, "Distributed team formation in urban disaster environments," in *2014 IEEE Symposium on Intelligent Agents (IA)*. IEEE, 2014, pp. 57–64.

[12] J. Capitan, M. T. Spaan, L. Merino, and A. Ollero, "Decentralized multi-robot cooperation with auctioned pomdps," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 650–671, 2013.

[13] B. P. Gerkey and M. J. Matarić, "Murdoch: Publish/subscribe task allocation for heterogeneous agents," in *Proceedings of the fourth international conference on Autonomous agents*. ACM, 2000, pp. 203–204.

[14] M. B. Dias and A. Stentz, "Traderbots: A market-based approach for resource, role, and task allocation in multirobot coordination," 2003.

[15] L. Luo, N. Chakraborty, and K. Sycara, "Competitive analysis of repeated greedy auction algorithm for online multi-robot task assignment." IEEE, 2012, pp. 4792–4799.

[16] P. MacAlpine, F. Barrera, and P. Stone, "Positioning to win: A dynamic role assignment and formation positioning system," in *RoboCup-2012: Robot Soccer World Cup XVI*, X. Chen, P. Stone, L. E. Sucar, and T. V. der Zant, Eds. Berlin: Springer Verlag, 2013.

[17] J. De Hoog, S. Cameron, and A. Visser, "Autonomous multi-robot exploration in communication-limited environments," in *Proc. of the 11th Conference Towards Autonomous Robotic Systems*. University of Plymouth, School of Computing and Mathematics, 2010.

[18] F. Lyardet, D. W. Szeto, and E. Aitenbichler, "Context-aware indoor navigation," in *Ambient Intelligence*. Springer, 2008, pp. 290–307.

[19] W. H. Mou, M. F. Chang, C. K. Liao, Y. H. Hsu, S. H. Tseng, and L. C. Fu, "Context-aware assisted interactive robotic walker for parkinson's disease patients," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 329–334.

[20] M. Beetz, T. Arbuckle, M. Bennewitz, W. Burgard, A. Cremers, D. Fox, H. Grosskreutz, D. Hahnel, and D. Schulz, "Integrated plan-based control of autonomous service robots in human environments," *IEEE Intelligent Systems*, vol. 16, no. 5, pp. 56–65, 2001.

[21] G. Kaminka, I. Frenkel *et al.*, "Integration of coordination mechanisms in the bite multi-robot architecture," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 2859–2866.

[22] J. Reich, V. Misra, D. Rubenstein, and G. Zussman, "Connectivity maintenance in mobile wireless networks via constrained mobility," *Selected Areas in Communications, IEEE Journal on*, vol. 30, no. 5, pp. 935–950, 2012.

[23] G. Hollinger and S. Singh, "Multi-robot coordination with periodic connectivity," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 4457–4462.

[24] G. Hollinger, S. Singh, J. Djugash, and A. Kehagias, "Efficient multi-robot search for a moving target," *The International Journal of Robotics Research*, vol. 28, no. 2, pp. 201–219, 2009.

[25] C. Geyer, "Active target search from uavs in urban environments," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*. IEEE, 2008, pp. 2366–2371.